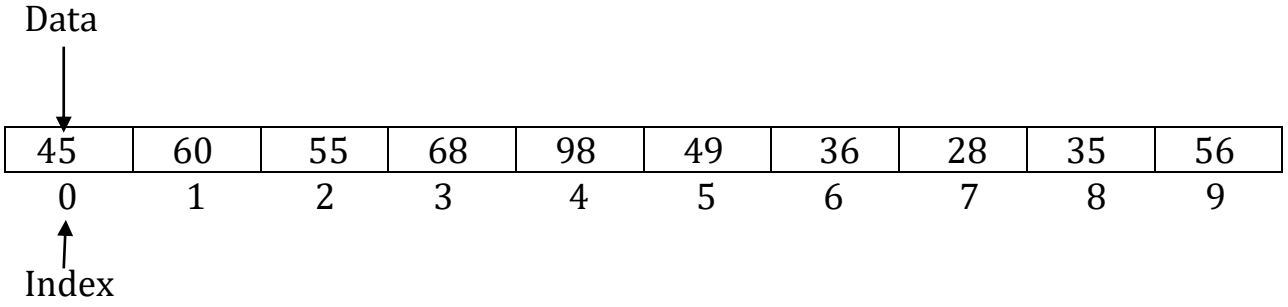
Topic : Array in C Programming

1. Creation of Array:

```
int x[10];
```



An array is a continuous memory location which store similar type of data. It has ability to use a single name to represent a collection of data items that enables us to develop concise and efficient program. It is a linear data structure, which is used to represent various data structure like STACK, QUEUE, and TREES etc.

Some examples where the concept an array can be used, such as:

1. List of employees in an organization.
2. List of products and their cost sold by a store.
3. Test of scores of a class of students.
4. List of customers and their telephone numbers.
5. Table of daily rainfall data etc.

2. Types of Array :

We can use arrays to represent not only simple lists of values but also tables of data in two, three or more dimensions. So, there are three types of arrays. These are:

1. One-dimensional Array.
2. Two-dimensional Array.
3. Multi-dimensional Array.

1. One-Dimensional Array:

A list of items can be given in one variable name using only one subscript, such variable is called a single-subscripted variable or a One-dimensional Array.

1.1 Declaration of One-Dimensional Arrays:

The general form of One-dimensional Array declaration is:

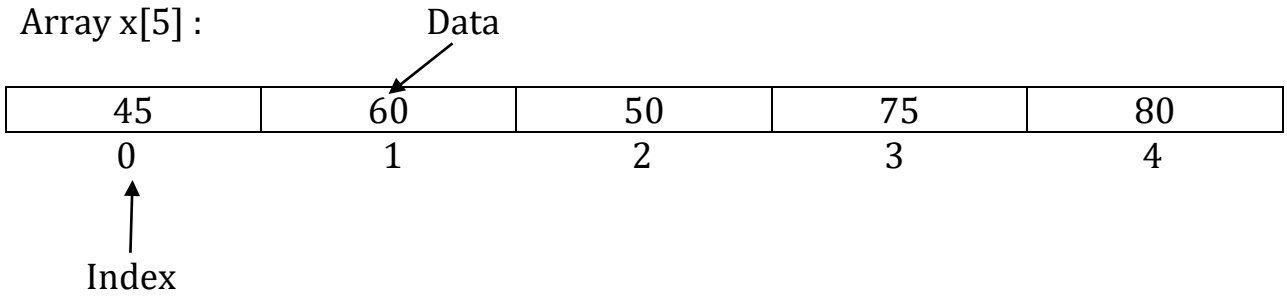
Data_type Variable_name[Size];

The data type specifies the type of element that will be contained in the array, such as char, int, float etc. the variable_name indicates the array name and the size specifies the maximum number of elements that can be stored inside the array.

Example:

int x[5];

Computer reserved five continuous memory locations



1.2 Initialization of One-Dimensional Array:

After an array is declared, its elements must be initialized, otherwise they will contain garbage value. An array can be initialized at either of the following stages:

1. At Compile Time.
2. At Run Time.

1. Compile Time Initialization:

The general form of compile time initialization is:

Data_type Variable_name[Size]={List of values};

Some important facts of compile time initialization are:

1. Int x[5]={10,15,20}
   Here remaining two elements will be set to zero automatically.
2. Int x[]= {4,7,8,2,9};
   Here, compiler allocates enough space for all initialized elements.
3. Char x[5]={'G','O','O','D','\0'};
   Character always written between single quote and last character is NULL .
4. Char name[]="BOY";
   It is also valid.
5. Char x[5]={'A'};
   It will initialize the first element to 'A' and the remaining to NULL.
6. Int x[3]={10,20,30,50,60};
   Here, the compiler will produce an error. This statement will not work. It is illegal in C.

2. Run Time Initialization:

A. An array can be explicitly initialized at run time. This approach is usually applied for initializing large arrays.

Example:

```
……..
……..
for (i=0;i<100;i++)
{
    Add[i]=0.0;          /* Assignment Statement*/
}
……..
……..
```

B. We can also use a read function such as scanf() function to initialize an array.

Example:

```
int x[5];
for(i=0;i<5;i++)
{
    scanf{"%d",&x[i]);
}
```

/* Q1: Write a C program which will read 10 numbers of elements and store into an array and also display that numbers. */

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
	int  x[10], i;
	clrscr();
	printf("Enter any 10 numbers");
	for(i=0;i<10;i++)
	{
		scanf("%d",&x[i]);
	}
	for(i=0;i<10;i++)
	{
		printf("%d",x[i]);
	}
getch();
}
```

/* Q2: Write a C program to find out the largest element in the array. */

Source Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int  x[10];
        int  i, large;
        clrscr();
        printf("Enter any 10 numbers");
        for(i=0;i<10;i++)
        {
                scanf("%d",&x[i]);
        }
        large=x[0];
        for(i=1;i<10;i++)
        {
                If(x[i]>large)
                large=x[i];
        }
        printf("%d",large);
        getch();
}
```